VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTSE@ 2019, ISSN(e): 2309-6519; ISSN(p): 2411-6327
Volume 14, Number 2, May-August , 2019 pp. 25—32

A MODERN APPROACHES FOR MODELING TIME-VARYING DATABASE MODELS

NASHWAN ALROMEMA[1*], FAHAD ALOTAIBI[2]
[1]Department of Computer Sciences, Faculty of Computing and Information Technology-Rabigh, King Abdulaziz University, Jeddah Saudi Arabia
[2]Department of Information System, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah Saudi Arabia
*[1]Email: nashwan.alromema@gmail.com;[2]Email: fmmalotaibi@kau.edu.sa

ABSTRACT. *Time-varying data models store data related to time instances and offer different types of timestamping. These modeling approaches are considered as one of the most important parts of many database applications like metrological, banking, biomedical, accounting, scheduling, reservation systems, sensor based systems, real-time monitoring applications and applications involving maintenance of huge records. This research work introduces the state-of-the-art modeling approaches of Time-varying data. Furthermore we will show how to represent a running example using different approaches and give a comparison study of storage, and the ease of use of each model.*
***Keywords**:* Time-varying Data, Relational Data Model, interval-based timestamping, valid-time data model, transaction time data model.

1. **Introduction.** Temporal database is the most popular database modeling technique that plays a key role in managing time-reference data that is important to almost every computer database application. In database technology, the most common techniques for database modeling are relational database, object-oriented database, spatial database and temporal database [4]-[6], [14]-[16]. These modeling techniques are defined in literatures as the following: (1) Relational Data Model (RDM) that has been implemented in a large number of commercial DBMS over the last three decades. RDBMS is very powerful because it is based on solid mathematical foundation [3]-[6], [28]-[32]. Relational data model stores the data that are considered to be valid at time instance now. (2) Object-Oriented Data Model (OODM) stores data about entities in objects, the objects that are of the same type are called collections (classes), and thus an object-oriented database contains a set of collections. The main goal of object-oriented database is to represent and model the data and relationship of the real world in natural manner [6], [30]. OODM is a type of database systems that apply the concepts of object orientation. The most common types are the object-relational database and temporal object-oriented database. The object orientation of time-varying data has been also considered in the research area of temporal database by incorporating time dimension to the data object [6], [19]. (3) Spatial Data Model, Spatial Data Model (SDM) models and stores a collection of space related data. Spatial approach provides a data structure and operations related to geometric or geographical data. The 2-Dimention of the Earth's surface and the 3-dimention representation of the chain of proton molecules are an example of spatial database. The management of historical changes of spatial data with respect to time rise to a new research area in which temporal database and spatial database are combined to form a research area spatio-temporal databases [16]. (4)Temporal Data Model, Temporal Data Model (TDM) stores data related to time instances. It offers temporal data types, kinds of time and temporal statement. It also stores information related to past, present and future time, for example, the change of employee's salary within an organization. Temporal database is one of the most important parts of many database applications like metrological database systems, banking, biomedical, accounting, scheduling, reservation systems, sensor based systems, real-time monitoring applications and applications involving maintenance of huge records [2], [10]-[12]. In the last four decades hundred researchers have produced many of research papers focusing on modeling temporal database by incorporating the concepts of temporal data model [4], [10]-[16], [19], [30]. A growing interest in temporal databases in many application domains has resulted in many books and research papers. The earliest dedicated to temporal databases was by [23]-[27] followed by others [1], [4], [14], [21] who addressed issues in handling time-varying data. Most of these publications covered different techniques for modeling, implementing, and developing temporal database. In this paper we provide some of the state-of-the-art modeling and representation approaches of Temporal Database Model (TDM) and investigates with discussion the limitations of these approaches.

2. Time-Varying TDM (Proof of concept). Temporal database model plays an important role in managing time-varying data. TDM can be incorporated with other database modeling techniques to provide expressive and efficient ways for representing and querying different type of time-varying data, these data can be relational data, object-oriented data, or data related to space. TDM can be incorporated with relational data model to model time-varying relational database, it can be incorporated with object-oriented database to model temporal object-oriented database, and finally, it can be incorporated with spatial database for modeling space related data (spatio-temporal database). Temporal data model represents the core of other data model; this is why most of database researchers focus on such research area. Figure 2.1 shows the number of published research papers in temporal database from the year of 2004 until 2016.

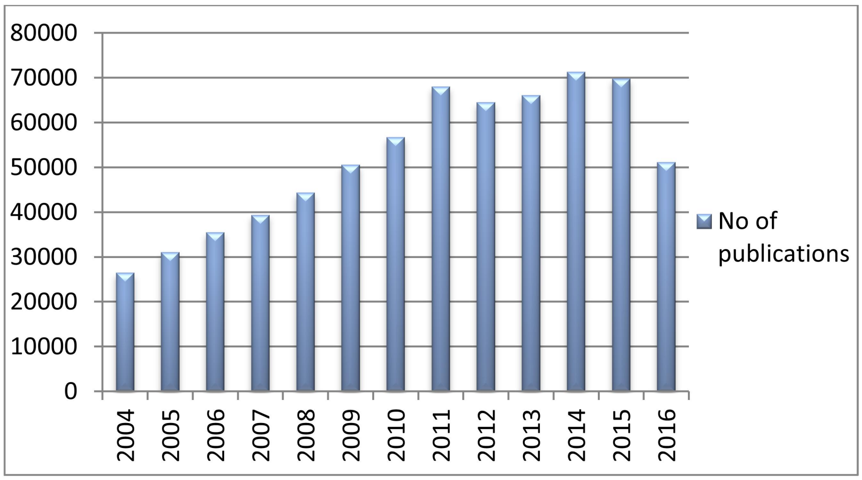

Figure-1: The publication items in temporal database (2004-2016)

This paper concentrates on the study of temporal database models and introduces an extended temporal relational data model that is based on schema extension approach discussed by [12], [21]-[22]. In the next section, the basic concepts and terminologies of temporal data model are introduced. as shown in Figure 2. The official definition of temporal database as stated in [9] is database with time supports. The important issues for modeling TDB are concentrated on a concepts as illustrated in Figure 2 and as stated in [21]. These concepts have a powerful guidance to think and represent/reasoning time-varying data. This paper emphasizes the way of thinking about time-varying data, shows with examples how to extend relational data model (using schema extension approach discussed by [22], [12], and maps these concepts in standard SQL. In the subsequent sections, temporal concepts and terminologies will be outlined.

3. The Running Example. Conventional Relation Model (CRM) is the most common relational data model that was first introduced by Codd in IBM Research in 1979 [1]. The components of this model contain a set of attributes, and does not have temporal supports [6]. The structure of the relation in CRM is defined as $R \subset A_1 \times A_2 \times A_3 \ldots\ldots\ldots\ldots\ldots \times A_n$, where $A_1, A_2, A_3, \ldots\ldots\ldots\ldots\ldots A_n$ and n > 0 are the atomic set of attributes that construct R. In TDB these attributes can be categorized into key attributes, time-invariant attributes, time-varying attributes, and finally the timestamp attributes. We assume key attributes are A_K, time-invariant attributes (unchangeable) are A_U, time- varying attributes (changeable) are A_C, the timestamp attributes are A_T. The schema intension of R can be redefined as $R(A_K, A_U, A_C, A_T)$ and each subset attribute contains a set of attributes from the same category i.e. for key attributes, $A_K = \{A_{K_1}, A_{K_2} \ldots\ldots\ldots A_{K_i}\}$, for time-invariant attributes $A_U = \{A_{U_1}, A_{U_2} \ldots\ldots\ldots A_{U_j}\}$, for time- varying attributes $A_C = \{A_{C_1}, A_{C_2} \ldots\ldots\ldots A_{C_m}\}$, and for timestamp attributes $A_T = \{A_{T_1}, A_{T_2} \ldots\ldots\ldots A_{T_z}\}$. The degree of R will be $(i + j + m + z)$. An example of a relation schema for a relation which describe students in the university is like $\text{Student(St\_id, SSN, B\_date, Birth\_country, Phone\_no, Major, GPA)}$, in this relation schema the set of key attributes is $A_K = \{\text{St\_id, SSN}\}$, the set of time- invariant attributes is $A_U = \{\text{B\_date, Counrty\_Birth}\}$ the set of time- varying attributes is $A_C = \{\text{phone\_no, Major, GPA}\}$. For the timestamp attributes, there are many temporal data models, namely valid-time, transaction-time, bitemporal-time, and lifespan time, in this study we will consider

the valid time aspects for attributes object by adding Valid-Start-Time(VST) and Valid-End-Tim(VET), and lifespan time aspects by adding Lifespan-Start-Time(LSST) and Lifespan-End-Tim(LSET) for entity and relationship objects. The relation schema of the student relation can be extended to support valid-time aspects of time by adding valid-time. Student relation will be redefined as:
$Student_T$ (St_id, SSN, B_date, Birth_country, Phone_no, Major, GPA, VST, VET, LSST, LSET)
For making the idea more clear, let us represent the student relation in tabular form as shown in next sections. In the following sections we will represent student relation in different approaches of temporal database models.

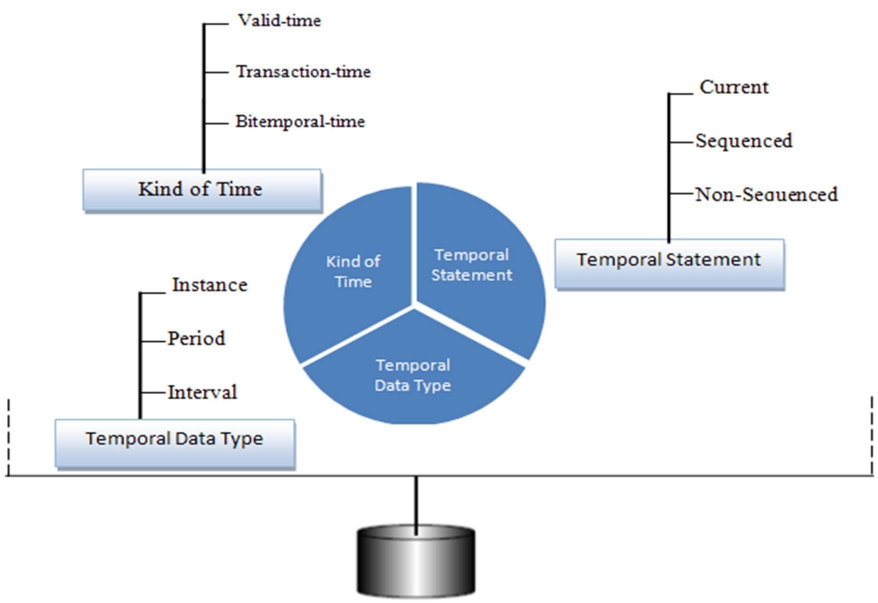

Figure-2: The orthogonal Concepts of Temporal Database Model

3.1 Temporal Query Language (TQuel). Snodgrass Model, TQuel, uses 1NF Tuple Timestamped Representation Scheme [20],[21]. This model incorporate time by adding additional attributes namely: Starting valid-time chronon (Vs) and ending valid-time chronon (Ve) are atomic-valued timestamp attributes as shown in **Figure-3**.

Student$_T$

St_id	*SSN*	*Name*	*B_Date*	*Birth_C*	*P_no*	*Major*	*GPA*	V_s	V_e
200101	19012	Jake	1/1/1990	UK	1232060	CS	3.2	5	13
200101	19012	Jake	1/1/1990	UK	1232060	CS	4.5	13	16
200101	19012	Jake	1/1/1990	UK	1232060	Math	4.5	16	21
200102	19012	Kate	1/1/1990	CA	9638222	Phys	3.2	7	23
200101	19012	Jake	1/1/1990	UK	1302222	Math	4.5	21	Now
200101	19012	Kate	1/1/1990	CA	9638222	Phys	3.5	23	Now

Figure-3: An example of Snodgrass Tuple Timestamped Representation Scheme.

3.2 Backlog-Based Representation Scheme (BLRS). Jensen's model [10], BLRS, uses bitemporal 1NF relations, The difference between this model and the previous one (Sondgrass model) is that tuples in backlogs are never updated, i.e., backlogs are append-only. Figure 4 shows an example of Jensen's model representation, where the attributes Vs and Ve store starting and ending valid time chronons respectively. Op attribute is used to indicate the Tuples change requests, insertion requests or deletion requests, as indicated by the values, I, and D, of attribute Op.

Student$_T$

St_id	***SSN***	***Name***	***B_Date***	***Birth_C***	***P_no***	***Major***	***GPA***	***V_s***	***V_e***	***Op***
200101	19012	Jake	1/1/1990	UK	1232060	CS	3.2	5	13	I
200101	19012	Jake	1/1/1990	UK	1232060	CS	4.5	13	16	D
200101	19012	Jake	1/1/1990	UK	1232060	Math	4.5	16	21	D
200102	19012	Kate	1/1/1990	CA	9638222	Phys	3.2	7	23	D
200101	19012	Jake	1/1/1990	UK	1302222	Math	4.5	21	Now	I
200101	19012	Kate	1/1/1990	CA	9638222	Phys	3.5	23	Now	I

Figure-4: An example of Jensen's backlog Relation Representation

3.3 Parametric Model (PRM). Gadia's, PRM, Model uses N1NF attribute value Timestamped Representation Scheme[7], [8]. A tuple is composed of a valid-time interval [Vs, Ve], and attribute value. Figure 5 shows how Student temporal data can be represented by Gadia's Model.

St_id	***SSN***	***Name***	***B_Date***	***Birth_C***	***P_no***	***Major***	***GPA***
200101	19012	Jake	1/1/1990	UK	1232060 [5 21] 1302222 [21 now]	CS [5 16] Mat [16 now]	3.2 [5 13] 4.5 [13 now]
200102	19012	Kate	1/1/1990	CA	9638222 [7 now]	Phys [7 now]	3.2 [7 23] 3.5 [23 now]

Figure-5: An example of Gadia's Model Representation.

3.4 Historical Data Model (HDM). Novikov and Gorshkova proposed a technique/data model, HDM, for implementing TDB in conventional DBMS using 1NF data model[17]. An example of representing the student relation in this model is shown in **Figure** 6.

Student

St_id	***SSN***	***Name***	***B_Date***	***Birth_C***	***P_no***	***Major***	***GPA***
200101	19012	Jake	1/1/1990	UK	1302222	Math	4.5
200102	19012	Kate	1/1/1990	CA	9638222	Phys	3.5

H_Student$_T$

St_id	***SSN***	***Name***	***B_Date***	***Birth_C***	***P_no***	***Major***	***GPA***	***V_s***	***V_e***
200101	19012	Jake	1/1/1990	UK	1232060	CS	3.2	5	13
200101	19012	Jake	1/1/1990	UK	1232060	CS	4.5	13	16
200101	19012	Jake	1/1/1990	UK	1232060	Math	4.5	16	21
200102	19012	Kate	1/1/1990	CA	9638222	Phys	3.2	7	23

Figure-6: Student Temporal relations in HDM model.

3.5 Tuple Timestamping Multiple Relations(TTMR). This approach based on decomposing the temporal relation as one relation for each time varying attribute, and one relation for time-invariant attributes. Temporal Relational Model in [5],[12], [21], [24], and [31] are using this approach. The time representation of the stored facts can be interval-based timestamps using valid-time model as shown in Figure 7.

Student

St_id	***SSN***	***Name***	***B_Date***	***Birth_C***
200101	19012	Jake	1/1/1990	UK
200101	19012	Jake	1/1/1990	UK
200101	19012	Jake	1/1/1990	UK
200102	19012	Kate	1/1/1990	CA
200101	19012	Jake	1/1/1990	UK
200102	19012	Kate	1/1/1990	CA

St_GPA

St_id	***GPA***	***V_s***	***V_e***
200101	3.2	5	13
200101	4.5	13	Now

200102	3.2	7	23
200102	3.5	23	Now

St_P_no

St_id	P_no	V_s	V_e
200101	1232060	5	21
200102	9638222	7	Now
200101	1302222	21	Now

St_Major

St_id	Major	V_s	V_e
200101	CS	5	16
200101	Math	16	Now
200102	Phys	7	Now

Figure-7: Student relation in TTMR data Model.

3.6 Tuple Timestamp Historical Relation (TTHR). This Temporal relational data model proposed in [9], [29], the schema of relation in TTHR is defined as two relations namely, R and R_T such that $R = (A_K, A_U, A_C)$, and $R_T = (A_K, Att\_index, Att\_value, A_T)$. R_T is an auxiliary relation schema as shown in Figure .

Student 1 2 3 4 5 6 7

St_id	SSN	Name	B_Date	Birth_C	P_no	Major	GPA
200101	19012	Jake	1/1/1990	UK	1302222	Math	4.5
200102	19012	Kate	1/1/1990	CA	9638222	Phys	3.5

Student$_T$

St_id	Att_index	Att_Value	V_s	V_e
200101	7	3.2	5	13
200101	6	CS	5	16
200101	5	1232060	5	21
200102	7	3.2	7	23

Figure-8: Student Temporal database relation in TTHR.

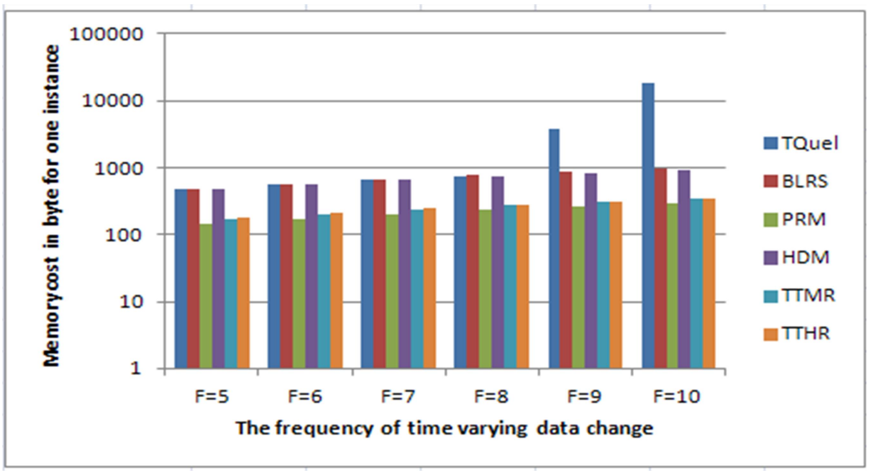

Figure-9: The memory storage cost in byte for each model in different frequency and U = 43 byte

4. Discussions and Result Analysis. The Temporal relational data model of student relation has been represented by 6 different data model, namely, TQuel, BLRS, PRM, HDM, TTMR, TTHR. For calculating the memory storage for each model we assume that the memory storage needed for key attributes is $\mathrm{Cost}(A_K) = K$ byte, for time-invariant attributes $\mathrm{Cost}(A_U) = U$ byte, for time-varying attributes $\mathrm{Cost}(A_C) = C$ byte, and for timestamping attributes $\mathrm{Cost}(A_T) = T$ byte. The cost of representing the student data in each model has two directions, the first one is the snapshot data for data instant i.e. the data for the student object being inserted for the first time, the second direction is about the frequency (*f*) of time-varying data change. In some applications this factor -(*f*)- is very high, in which we should be choose the model that doesn't cost much of memory storage. In **Figure 9** the cost of memory storage for student data represented in temporal data model calculated for 6 scenarios, in which *f* varies from 5 to 10 and by having $\mathrm{Cost}(A_U) = U = 43$ byte for single instance, consequentially the memory storage has been calculated for each model.

In the second scenario, we fixed all the parameters and increase the cost of time-invariant attributes as shown in **Figure 10,** the cost of memory storage for student data represented in temporal data model calculated for *f* varies from 5 to 10 and by having $\mathrm{Cost}(A_U) = U = 120$ byte for single instance, consequentially the memory storage will be calculated for each model as in **Figure 10**.

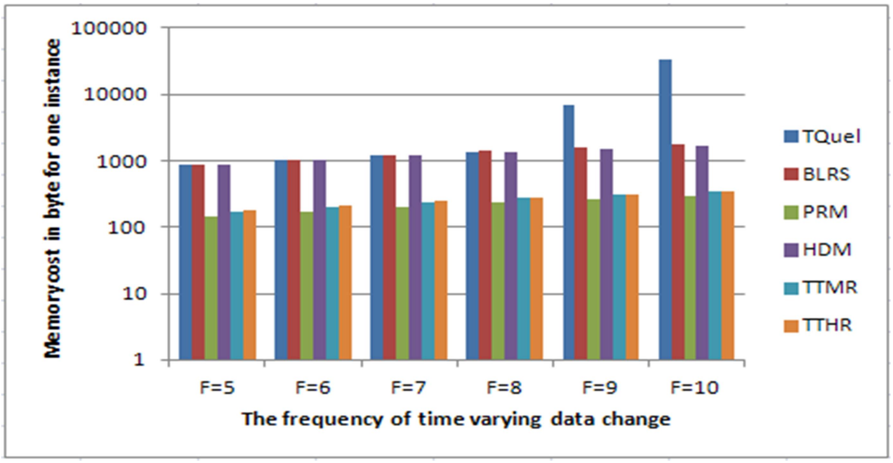

Figure-10: The memory storage cost in byte for each model in different frequency and U = 120 byte

Conclusion. This research has focused on outlining the state-of-the-art modeling approaches of temporal database models. Six (6) data models, namely, TQuel, BLRS, PRM, HDM, TTMR, and TTHR, have been discussed and used to represent a simple example of time-varying-data. Some of these models are in 1NF while others are in N1NF model. TTHR is an example of 1NF models, it has been proposed in [9], which is a consistent extension of CRM and has the expressive power to be represented in the reference temporal data model BCDM and the main temporal data models in literature. Some of the models discussed in this paper does not impose many constraints on the database operation, thereby making it viable to be adopted in a broad range of functioning database applications. These models have been compared in terms of memory storage for six (6) different frequency of time and by varying the cost of time-invariant attributes into two values. The trends in many approach of modeling temporal data models are focusing on reducing the memory storage and query processing time taking into consideration the full usage of the existing database technology without any changes. Unlike other approaches, it do not either introduce needless redundancy of data, nor cost a lot for querying the current snapshot data thereby making use of only a single relation for recording the valid time changed data.

ACKNOWLEDGMENT

This paper was funded by the Deanship of Scientific Research (DSR) under project number (G:1708-611-1440), King Abdulaziz University. The authors, therefore, acknowledge with thanks to DSR's technical and financial support.

REFERENCES

[1] Date, C. J., Darwen, H., & Lorentzos, N. (2002). Temporal data & the relational model. Elsevier.

[2] Anselma, L., Bottrighi, A., Montani, S., & Terenziani, P. (2013). Extending BCDM to cope with proposals and evaluations of updates. IEEE Transactions on Knowledge and Data Engineering, 25(3), 556-570.

[3] Ariav, G. (1986). A temporally oriented data model. ACM Transactions on Database Systems (TODS), 11(4), 499-527.
[4] Arora, S. (2015, September). A comparative study on temporal database models: A survey. In Advanced Computing and Communication (ISACC), 2015 International Symposium on (pp. 161-167). IEEE..
[5] Atay, C. (2016). An attribute or tuple timestamping in bitemporal relational databases. Turkish Journal of Electrical Engineering & Computer Sciences, 24(5), 4305-4321.
[6] Saxena, V., Pratap, A. (2013). Performance comparison between relational and object-oriented databases. International Journal of Computer Applications, 71(22) doi:http://dx.doi.org/10.5120/12616-8275.
[7] Garani, G. (2003). A temporal database model using nested relations (Doctoral dissertation, BIRKBECK COLLEGE).
[8] Garani, G. A generalised relational data model. International Journal of Computer Systems Science and Engineering 2007; 11, 43-59.
[9] AL-romema, N. (2010). Memory storage issues of temporal database applications on relational database management systems. Journal of Computer Science, 6(3).
[10] Jensen, C. S., Snodgrass, R. T., & Soo, M. D(1995). The tsql2 data model. Springer US; (pp. 157-240).
[11] Kumar, S. and Rishi, R. (2016a). Retrieval of Meteorological Data using Temporal Data Modeling. Indian Journal of Science and Technology, 9(37).
[12] Kvet, M., Matiako, K. and Kvet, M.(2014). Transaction management in fully temporal system. In Computer Modelling and Simulation (UKSim), UKSim-AMSS 16th International Conference on (pp. 148-153). IEEE.
[13] McKenzie Jr., and Snodgrass, R. T.(1991). Evaluation of relational algebras incorporating the time dimension in databases. ACM Computing Surveys (CSUR); 23(4), 501-543.
[14] Terenziani, P. (2016a). Irregular indeterminate repeated facts in Temporal Relational Databases. IEEE Transactions on Knowledge and Data Engineering, 28(4), pp.1075-1079.
[15] Noh, S.Y., and Gadia, S. K. (2008). Benchmarking temporal database models with interval-based and temporal element-based timestamping. J SYST SOFTWARE; 81(11):1931–1943.
[16] Viqueira, J.R.R. and Lorentzos, N.A. (2007). SQL extension for spatio-temporal data. The VLDB Journal—The International Journal on Very Large Data Bases, 16(2), pp.179-200.
[17] Novikov, B. A., and Gorshkova, E. A. (2008). Temporal databases: From theory to applications. PROGRAM COMPUT SOFT+; 34(1), 1-6.
[18] Sarda, N. L. (1990). Algebra and query language for a historical data model. The Computer Journal; 33(1), 11-18.
[19] Chau, V.T.N. and Chittayasothorn, S. (2007). A temporal compatible object relational database system. In Proceedings 2007 IEEE SoutheastCon (pp. 93-98). IEEE.
[20] Snodgrass, R. (June 1987). The Temporal Query Language TQUEL. ACM T DATABASE SYST 1987; 12(2):247.
[21] Snodgrass, R. (2000). Developing Time-Oriented Database Applications in SQL, 1st edition, Morgan Kaufmann Publishers, Inc., San Francisco.
[22] Steiner, A. A Generalization Approach to Temporal Data Models and their Implementations. A dissertation submitted to the SWISS FEDERAL INSTITUTE OF TECHNOLOGY, ZURICH, 1998.
[23] Tansel, A. Adding time dimension to relational model and extending relational algebra. INFORM SYST 1986; 11(4), 343-355.
[24] Tansel, A., and Snodgrass, R. (1993).Temporal databases: Theory, design and implementation. Redwood City, CA: Benjamin Cummings.
[25] Tansel, A. (2004) .Temporal data modeling and integrity constraints in relational databases. In Computer and Information Sciences-ISCIS; (pp. 459-469). Springer Berlin Heidelberg.
[26] Tansel, A. (2004). On handling time-varying data in the relational data model. INFORM SOFTWARE TECH; 46(2), 119-126.
[27] Tansel, A. (2013).Modeling and Querying Temporal Data, Idea Group Inc 2006;
[28] Terenziani, P. Coping with events in temporal relational databases. IEEE TKNOWL DATA EN, IEEE Transactions; 25(5), 1181-1185.
[29] Alromema, N., Rahim, M.S.M. and Albidewi, I., (2016). Performance Evaluation of Attribute and Tuple Timestamping In Temporal Relational Database. International Journal of Computer Science and Information Security (IJCSIS), 14(09).
[30] Zhang, Q. (2001). Object-oriented database systems in manufacturing: selection and applications. Industrial Management and Data Systems, Vol. 101 Iss: 3, pp.97–105.

[31] Navathe, S. B., and Ahmed, R. (1989). A temporal relational model and a query language. Information Sciences, 49(1), 147-175.
[32] Petkovic, D. (2016). Temporal Data in Relational Database Systems: A Comparison. In New Advances in Information Systems and Technologies (pp. 13-23). Springer International Publishing.